2004

Forecasting hospital bed availability using computer simulation and neural networks

Matthew J. Daniels

Rochester Institute of Technology

FORECASTING HOSPITAL BED AVAILABILITY USING COMPUTER SIMULATION AND NEURAL NETWORKS

A Thesis

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Industrial Engineering in the

Department of Industrial & Systems Engineering

Kate Gleason College of Engineering

By Matthew J. Daniels

B.S., Industrial Engineering, 2003

August 2004

Thesis/Dissertation Author Permission Statement

Title of thesis or dissertation: Forecasting Hospital Bed Availability Using Computer Simulation and Neural Networks

Name of author: Matthew J. Daniels
Degree: M.S., Industrial Engineering
Program: Industrial and Systems Engineering
College: Kate Gleason College of Engineering

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Print Reproduction Permission Granted:

I, Matthew J. Daniels, hereby **grant permission** to the Rochester Institute Technology to reproduce my print thesis or dissertation in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author: Matthew J. Daniels Date: 2/15/04

Print Reproduction Permission Denied:

I, _____, hereby **deny permission** to the RIT Library of the Rochester Institute of Technology to reproduce my print thesis or dissertation in whole or in part.

Signature of Author: _____ Date: _____

Inclusion in the RIT Digital Media Library Electronic Thesis & Dissertation (ETD) Archive

I, _____, additionally grant to the Rochester Institute of Technology Digital Media Library (RIT DML) the non-exclusive license to archive and provide electronic access to my thesis or dissertation in whole or in part in all forms of media in perpetuity.

I understand that my work, in addition to its bibliographic record and abstract, will be available to the world-wide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I am aware that the Rochester Institute of Technology does not require registration of copyright for ETDs.

I hereby certify that, if appropriate, I have obtained and attached written permission statements from the owners of each third party copyrighted matter to be included in my thesis or dissertation. I certify that the version I submitted is the same as that approved by my committee.

Signature of Author: _____ Date: _____

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

KATE GLEASON COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Matthew J. Daniels has been examined and approved by the thesis committee as satisfactory for the thesis requirement for the Master of Science degree

Approved by:

Michael E. Kuhl

Dr. Michael E. Kuhl, Advisor

Moises Sudit

Dr. Moises Sudit

Elisabeth Hager

Dr. Elisabeth Hager

Abstract

The success of hospitals in treating patients and staying in business relies on their efficient use of resources. In particular, the utilization of hospital beds is a critical concern, since over-crowding will result in delays or transfers of patients, and under-utilization will result in lost opportunity to treat patients and generate profit. To this end, hospital decision makers must have reliable forecasts of patient demand and bed availability. The objective of this thesis was to create a general method to forecast the availability of hospital beds in the short term, up to 2 days into the future. Specifically, this thesis employed a computer simulation model of the hospital and a tirne-dependent neural network to learn from the simulated model and forecast the availability of beds. The computer simulation model was found to be well suited to the task of describing a general hospital system and creating training data for a neural network. The neural network was found to provide accurate performance in predicting bed availability in the short term. The network incorporated the effect of tirne explicitly to capture the non-stationary behavior of hospital systems. These findings have a number of implications that will be discussed.

Table of Contents

List of Figures

List of Tables

1. Introduction

Health care is possibly the most important industry in the United States for the application of process improvement techniques because of its sheer size, its current performance with respect to efficiency, and its central place in our society.

The enormous size of the US health care system is plainly evidenced by the dollars it consumes, the number of patients it serves, and the number of people it employs. In terms of health care costs, total expenditures and average charges per hospital stay are two measures that illustrate the size of the US health care system: national health care expenditures were $1.4 trillion in 2001 (Centers for Medicare and Medicaid Services, 2003), and the national average charges per hospital discharge in 2002 were over $17,000, up from $11,000 five years earlier (Agency for Healthcare Research and Quality, 2004). In terms of the number of people served by health care, the 2.46 million patients admitted to hospitals in New York state in 2002 and the more than 37 million patients nationwide in the same year attest to the size of this industry (Agency for Healthcare Research and Quality, 2004). Finally, in terms of employment in this sector, the Bureau of Labor Statistics (2004) reports that 14,188,000 people were employed in health services as of July 2004 – that is, nearly 11% of US employment. Health care is truly a system of gigantic proportions in the United States.

Beyond the size of the system, its widely reported inefficient practices contribute to the need to make systematic improvements. Alarmingly, health care costs are rising at a rate of approximately 9% per year (Centers for Medicare and Medicaid Services, 2003). One culprit in the high cost of health care is inefficiency in the development of pharmaceuticals: "On average,

of 5,000 new compounds tested in the laboratory, only 5 actually make it to clinical trials... and only one is approved for patient use. Elapsed time can be 12 to 15 years. Hence, the average cost of a new medicine is around $500 million" ("Drug-price program notes," 2000). As a further example, US health care providers are notorious for being behind the times with respect to information technology: the Joseph H. Kanter Family Foundation (2002) reports that savings on the order of $80 billion per year could be realized by making better use of the most current information technology available. Any system of this size, especially with the inefficiency indicated in the US health care system, is an important target for study and improvement.

In addition to the size and inefficiency of health care systems, however, the nature of their function in our society provides yet another motivation to work toward their improvement. As mentioned above, they provide a large source of employment; and they are not only used by a large volume of people in this society, they are used by people in every demographic and every segment of it. Further, these services are not mere conveniences or luxuries: the wellness of all people depends, at some point in their lives, on their access to and adequate service from a health care provider. So, in summary, the US health care system is a critical subject for improvement by scientific methods.

Indeed, many improvement efforts have been undertaken over the years. One significant, though not completely successful, effort toward cost containment has been the transition from fee-for service payment to managed care. The existing fee-for-service system paid a provider according to any services rendered to a patient. Managed care, in contrast, is a system that pays health care providers based on the patient's diagnosis, rather than on the amount of resources

consumed in treating that patient. By not providing additional financial incentive to use extra resources on individual patients, it is intended as a means to deter overuse of health care resources. Under this system, the only way to generate more revenue is by treating more patients, and therefore an important result of managed care has been a significant decrease in patients' lengths of stay in hospitals.

Given this incentive to treat as many patients as possible to maximize profit, it has become imperative to closely manage the utilization of hospital beds. Several tools are used by hospitals to manage their hospital bed utilization. One important example is utilization review, where case managers track and evaluate each patient's use of resources. Medical technology – better procedures, tools, and medicines – is a second example of an important tool for utilization management. A final example is the scheduling of patients and staff; this critical tool is used in hospitals to try to control the flow of patients.

The optimization of bed utilization, despite the efforts of utilization review and other techniques, is difficult to achieve. The high variability in demand for services and patients' lengths of stay prohibits simple forecasting of demand. Without this ability to forecast the hospital's capacity status into the future, one of two negative situations will arise. In the first scenario, the hospital reserves a number of beds for patients that may come in through an Emergency Department. If these beds go unfilled, the hospital loses out on potential revenue. The second situation is that in which the hospital fills (or schedules to fill) as many beds as are available, with little regard to emergency patients at a future time. This creates a situation of over-crowding, where patients must either be transferred to another facility or, if they are

scheduled patients, must be turned away and re-scheduled. This, in turn, creates dissatisfaction on the part of patients and results in increasing waiting lists for scheduled surgical procedures. It is not difficult for a hospital to maximize bed utilization without respect to the negative scenarios above. However, it is difficult – and necessary for the success of the hospital system – to time-dependently *optimize* bed utilization. Therefore, intelligent methods of forecasting bed availability are crucial to maintaining high utilization while not delaying or turning away patients in need.

One important measure for optimizing bed utilization and the hospital's success is the number of available beds at points in the near future. If many beds are held for potential emergency patients and only a few patients show up, then beds sit empty that could have been used for scheduled procedures. On the other hand, if not enough beds are held for potential emergency patients and many patients do arrive, then either scheduled patients must be sent away and re-scheduled, or else emergency patients will have to be sent for treatment at a different hospital.

This proposal will describe the approach to a new solution this important problem. By forecasting future bed availability more accurately than current methods allow, hospitals will realize improvements in staffing accuracy and reductions in patients turned away upon arrival. Further, waiting lists for scheduled procedures will diminish as patients are treated on their first visit rather than being re-scheduled. The results of this improvement will be immediately felt. Better information about bed availability will allow scheduling of specialized nurses around the types of patients that are likely to be present, while potentially saving the cost of many shifts by

knowing busy and idle times in advance. Moreover, consider the impact on hospital revenue of optimized bed utilization. As cited previously, the national average charges per hospital discharge in 2002 were $17,000 (Agency for Healthcare Research and Quality, 2004). Maintaining the same levels of service to emergency patients while being able to accommodate just ten more discharges per year, for example, would generate additional revenue on the order of $170,000 per year.

This research addresses the identified need by developing a general method to accurately and quickly forecast bed availability up to 48 hours in advance. The general method uses a computer simulation model with a neural network forecasting system, in a two-phase approach, to provide an expected value and associated tolerance interval for each prediction. The computer simulation model provides an accurate picture of the processes and interactions in the real system. As such, it can also provide volumes of usable training data for the neural network. In the second phase of the method, the neural network forecasting system uses the simulated data for training, after which it can quickly generate predictions, and it can continually update its learning based on newly received actual values to remain accurate and relevant.

This paper is organized as follows: Section 2 contains the Problem Statement. Section 3 reviews literature related to forecasting, computer simulation, and neural networks, as well as the application of each in health care. Section 4 details the general method for creation of the forecasting system for the case of predicting one time step into the future, and the various experiments performed are described. Conclusions and recommendations for future work follow in Section 5.

2. Problem Statement

The problem addressed by this research is that of accurately forecasting the availability of hospital beds into the immediate future. The problem of managing bed utilization is important: overcrowded hospitals mean delaying, re-scheduling, or transferring to other hospitals the treatment of many patients; meanwhile, under-utilized beds mean a squandered opportunity to treat patients and generate revenue.

This research will apply a novel approach to this forecasting problem through the use of two phases. First, a typical hospital system will be modeled through the use of computer simulation. This simulation model will allow the study and understanding of the system being modeled. Specifically, factors most influencing bed availability and utilization will be examined. The model will create an understanding of arrival patterns of patients, treatment lengths, and current resource utilization in the system. This model will be verified and validated, to ensure that it accurately represents reality and can provide reliable outputs.

Second, the simulation model will be used to apply a neural network, to increase the efficiency of the solution on an everyday basis. This neural network phase will provide two key benefits in forecasting bed availability and, therefore, in optimizing bed utilization. The first benefit is increased speed over the use of the simulation model for everyday forecasting needs. While a simulation model of this complexity may take at least several minutes to achieve a forecast result, because it requires sufficient run time to build an acceptable confidence interval of the estimate, a neural network can achieve a result much more quickly, without sacrificing accuracy. The second key benefit of this approach is the inherent ability of neural networks to

adapt themselves to updated actual data; that is, as conditions change in the system, the neural network will be able to "learn" these changes and continue to produce accurate results. Its effectiveness will not be limited to a static situation of the system parameters.

The type of result that will be produced by the forecasting system is an estimate of the expected number of beds and an associated tolerance interval, for one to 48 hours into the future.

3. Literature Review

Two characteristics of the real world make forecasting an important subject for study: one is randomness, and the other is scarcity. With certain knowledge of future demand, or with unlimited resources to meet future demand, success could be guaranteed. However, since there is variability in demand for resources, and since resources to meet that demand are limited, success is based on the optimal use of resources to meet demands. Thus, forecasting is critical to success in the real world: optimal forecasts provide the opportunity for optimal usage of resources. As MacNiece (1961) succinctly writes, without forecasting "both short- and long-range planning rests on foundations much less substantial than sand" (p. 109).

This thesis addresses the problem of short-term forecasting of hospital bed availability; as such, this literature review will discuss how this problem has been addressed and what methods are available to arrive at a better solution. Traditionally, forecasting in health care usually considers long-term forecasts of resource utilization, rather than hour-to-hour forecasts. These methods are shown to be inadequate for use in the thesis problem. Other methods that are available for forecasting short-term bed utilization include Expert Systems and artificial neural networks. While neural networks require a large amount of data, and the determination of network structure and inputs can be problematic, they are shown to be ideal for addressing this problem when complemented by computer simulation models.

3.1 Long-Term Forecasting in Health Care

Most of the literature dealing with forecasting in health care systems is based on the long-term forecasting of bed utilization. These strategic forecasts are used to plan out a hospital's bed

and staff needs over long times, say over a period of months to years. In this section, some works in the long-term forecasting of health care systems are reviewed.

Côté and Tucker (2001) provide an excellent summary as a starting point for the practitioner of strategic forecasting of hospital demand: "Four common methods of forecasting are percent adjustment, 12-month moving average, trendline, and seasonalized forecast" (p. 54). The first two methods, percent adjustment and 12-month moving average, are simplistic forecasting techniques that assume no process changes. While this may be a valid assumption for some systems, the authors point out that these methods do not incorporate any of the seasonal effects that are inherent in monthly patient demand throughout a year. Where there are trends and spikes in the process that significantly affect the measure being forecasted, neither of the two methods will catch up to changing actual demand. Trend lines are the third forecasting technique described as common by Côté and Tucker, for predicting hospital demand in the medium and long term. This method is a simple case of a linear regression: a line is fitted to the observed data points in order to extract the expected trend. While this method is more sophisticated than the previous two methods, the effect of seasonality in the data is still not considered.

The final technique in this paper does account for seasonal trends explicitly and produce more realistic results than the other three techniques. The seasonalized forecast transforms the observed values into "deseasonalized" values by comparing each season's average to the overall average (for example, to say that month x is 10% above the average of all months). Using this information to relate all the data points back to a common baseline, a trend line is calculated, and forecast values from this regression line in a particular month are then inflated or deflated by that

month's seasonal factor. While this final method is often more appropriate for this task, based on the nature of the system, the authors issue a final caution with respect to all four methods: "Healthcare financial managers know that change is perhaps the only constant for the future. Therefore, knowledge of both intemal and external environments must be factored into final forecasts" to increase their practical usefulness and value (Côté and Tucker, 2001, p. 57).

Another paper on health care utilization forecasting in the long term is a contribution from MacStravic (2004). The author again emphasizes the inadequacy of simple models or measures in addressing complex problems: "I once 'proved' via an exponential trend line forecast, that in only 25 years, every person living in the State of Virginia would be a registered nurse.... Of course, no trend line, however large or small its goodness of fit, should be chosen for use in forecasting without understanding and having confidence in the continued and consistent action of whatever dynamics caused the past trend" (p. 11). The main deficiency of these simple models is precisely that inability to account for (and react to) the dynamics of the real health care world. This is the arena where more complex models are appropriate for forecasting.

In long-range utilization prediction in health care, models of increased complexity usually occur as extensions of the simple models described above. Woodruff's guidelines on bed need planning (2002) incorporate additional factors besides time-lagged values of demand itself. The recommendation is to include community-level measures, such as population change and market share, along with hospital-level variables like past discharge and length-of-stay data and resource usage trends. These variables are then used in a multi-step forecasting process to arrive

at estimates for the average daily census of patients and the peak census (when the hospital is most full). From this estimate, Woodruff recommends use of the normal probability distribution to determine a safe interval into which the needed number of beds should fall. Even for creation of a design-phase or strategic need forecast, the author acknowledges that the dynamics and variability of health care systems must be taken into account: "Predicting the variability of future demand is probably the most critical element of a sound bed need-planning model" (Woodruff, 2002, p. 2). Here, then, a more in-depth assignment of the causes of patient demand fluctuation is attempted, in order to accurately capture long-range bed need.

Continuing toward higher complexity is a final example from Myers and Green (2004). The authors incorporate the "incremental impact of medical advances" in their two-phase approach to forecasting utilization (p. 35). The first phase, again, is a traditional forecast as described in the foregoing papers. The authors' approach here attempts to add the effect of medical technology as one of the variables influencing utilization. The idea is, with medical advances come reduced patient lengths of stay and reduced chance that the same patient will return for the same procedure. This same idea could be generalized to any of a vast number of effects on health care utilization, including community immunization rates, climactic change, standard of living in the hospital's service area, and many more.

In summary, utilization forecasting in health care is often aimed at predicting utilization levels far into the future (months as opposed to days), as this impacts the hospital's construction/staffing priorities as well as its bottom line. For its importance, many of the commonly used models for these predictions are too simplistic to accurately capture changes that

can affect utilization. The sensitivity of hospitals to patient length of stay, for example, is highlighted in Myers and Green's paper (2004): "For a large system such as [the case study hospital], a 0.1-day change in LOS results in a 10- to 20-bed swing in required capacity" (p.37). Therefore, the models that have developed to address this complexity have added additional factors to these simple, basic methods.

3.2 Short-Term Forecasting in Health Care

While a number of instances are found that examine long-range utilization of hospital resources, fewer entries in the literature address short-term (i.e., hour-to-hour) forecasting in health care situations. The methods and scope found in the short-term utilization forecasting literature differ significantly from the relatively simple methods discussed above. In particular, the applications of forecasting seem to focus on single hospital units, most popularly the Emergency Department (ED). Further, the methods that are used are more complex and capable of handling the high variability and complexity inherent in hospitals. Two examples employing statistical methods and artificial neural networks are discussed in this section.

3.21 Time-series methods for short-term bed utilization forecasting

The statistical methods used in Jones, Joy, and Pearson's 2002 paper "Forecasting demand of emergency care" are notably different from the methods used in longer-range forecasting. The authors use an Auto-Regressive Inductive Moving Average (ARIMA) model to predict the daily number of occupied beds in a hospital due to emergency admissions. ARIMA is a statistical model that assumes that observed values of a process are the result of a "shock" to an overall process average, using the form

$$x_t = \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

(Jones, Joy, and Pearson, 2002, p. 297)

In the above equation, φ and θ are model parameters, and ε is an error term. The shock term can act as a type of seasonal modeling. This seasonal component is enhanced in the authors' work by use of the Seasonal ARIMA, or SARIMA, model to address recognized "strong seasonality" in the data (p. 298). This extended model adds two more terms to capture the history of observed values and shocks in past seasons. This method has promise as a forecasting method in this type of application: the authors achieve a root mean squared error of only 3% in their forecasts of the mean bed usage for emergency admissions. Nevertheless, this technique uses parameters that are fixed once they are determined; as such, these models are likely not suitable for highly dynamic and non-stationary systems.

3.22 Neural networks for mean E.D. wait time forecasting

Artificial neural networks (or simply neural networks) are another sophisticated forecasting method that has been applied to this specific problem by Kilmer, Smith, and Shuman (1997). A neural network (see full discussion of neural networks in Section 3.4 of this literature review) is a model that uses numerous, simple parallel processors to handle tasks such as pattern recognition, classification, machine control, and others, as well as for forecasting. The weights between the processors are changed based on feedback about the network's estimation errors, so that through training (rather than programming), the network "learns" the relationship of interest. As a result, the network is extremely flexible to change as the system changes, and once trained, forecast values can be generated almost instantly. The authors in this case use the neural network as a meta-model extension of a computer simulation model: the model creates data from the model system, which the network then "learns." The application, here again, is an Emergency Department, and the focus of the work is on estimating mean patient time in the E.D.

The authors begin by building a simulation model of the E.D., where patients arrive and are treated by various people and resources in the system. (See Section 3.5 of this literature review for a discussion of computer simulation). Once the model is built, the authors build parallel networks to estimate the mean and variance of patients' lengths of stay in the E.D. These networks are shown by the authors to perform at least as well as the simulation model. However, the inputs used in the model – the intensive care unit bed waiting time, general/surgical unit bed waiting time, lab service time, and X-ray service time – are not information that would be readily available in a real-world situation, and indeed if this information were known, a sophisticated model-network solution might not be necessary. Further, only a single hospital unit is modeled in the authors' work. Still, this methodology allows an outstanding richness and flexibility of modeling, and it is considered a superior approach to other techniques thus far discussed.

Research in health care forecasting has been progressing for some time, focused mostly on strategic capacity planning in hospitals. Where short-term bed availability forecasting is involved, the research appears still to be in early stages; however, the richness of the technology in use and the results obtained thus far are encouragements that these problems can be treated effectively. Because health care systems are complex, variable, and non-stationary, sophisticated methods like the meta-model computer simulation and neural network approach are most appropriate for addressing the forecasting of bed availability in the short term.

3.3 Other Methods for Short-Term Bed Utilization Forecasting

As highlighted in the above sections of this literature review, any method that will attempt to forecast hospital bed availability on a short-term basis, and remain relevant over time,

must have a combination of several features: it must produce quick results for real-time forecasting; it must account for hourly, daily, and other cyclical changes in the system; it must adapt as the system changes over time; and it must be capable of capturing enough of a hospital's complexity and inter-relatedness to adequately model measures based on the whole system's performance. Three explanatory methods are discussed in this section: regression, expert systems, and computer simulation.

3.31 Regression for forecasting

Regression is a general term encompassing a number of techniques, among which the common theme is that an output result can be explained by some combination of one or more input variables. Regression techniques vary greatly in sophistication. The basic form of regression is linear regression, which supposes that an output of interest arises as a linear function of one other variable. While this technique is simple, multiple linear regression is more sophisticated, modeling an output as a function of multiple variables; further, non-linear regression techniques allow such a model to transcend the linear restrictions of simpler forms of regression.

While these techniques are useful in many cases, and indeed a multiple regression model may be able to capture system behavior very well, the problem with these techniques is that they are static. For each technique, parameters of the model must be estimated that formalize the way in which inputs combine to create an output. Once these parameters are determined, however, they are fixed. Therefore, though regression is a useful technique that can explain a lot of system behavior, rather than just follow a time series of outputs, it does not meet the needs of the current thesis problem.

3.32 Expert systems for forecasting

Expert systems are a good example of the potential for using human expertise for problem solving. An expert system is a computer program that transforms experts' opinions into specific rules, so that the system is "sufficient to perform as a skillful and cost-effective consultant" (Bramer, 1982). The system takes in system inputs, and based on the values of those inputs and the rules encoded in the system, it generates a decision or predicted value. Thus, a good expert system should make the same decisions that a human expert would make, based on the same input information.

Expert systems have already been created to address issues in many fields, including knowledge engineering, medicine, chemistry, and others. One example of an expert system application in the clinical health care area is the HERMES system in the work of Bonfà, Maioli, Sarti, Milandri, and Dal Monte (1993). In this application, an expert system is built to determine, from various inputs, the prognosis of patients with potential liver diseases. A number of experts are interviewed to glean from them the process and rules by which prognoses are successfully made. Their process continued with encoding into a system, refinement of the rules (upon review with the experts), and finally implementation. Systems like this are capable of filling "the need to compare and evaluate forms of reasoning and to manage elements which cannot be rigidly schematized" (p. 240). A drawback of this method is that it, like regression techniques, is a static system once its rules have been specified. Further, from a practical standpoint, the time required to develop such a system (given the interviewing and rule encoding requirements) may prohibit its implementation in a reasonable time frame. The model discussed here, for example, was developed by the authors over a span of three years. In that span of time,

bed utilization dynamics can be expected to change significantly, rendering such a system obsolete by the time it is finished.

3.33 Computer simulation for forecasting

A third forecasting method that seeks to explain output behavior, rather than just follow its trend through time, is computer simulation. Simulation is a preferred method for learning about complex, inter-related systems, and as such, a valid and accurate computer simulation model is capable of producing reliable predictions of future performance of those systems.

While discussion of computer simulation is deferred to section 3.5 of this literature review, an example application of computer simulation shows its ability to capture the behavior of whole systems. Bagust, Place, and Posnett (1999) study hospital bed usage and the effect of emergency admissions on the system, through the use of computer simulation. By modeling a whole hospital and enabling the tracking of various measures of interest, the authors are able to reach significant conclusions regarding the availability and usage of beds: specifically, that when the hospital in this study was more than 85% occupied, the risk of running out of available beds became significant, with long-term impact. "Even a relatively low risk of failure can disrupt the operation of a hospital for a considerable time: at 85% mean occupancy, a hospital that runs out of beds for four days in a year may be disrupted for up to eight weeks in total" (p. 157). This example demonstrates the type of larger-context analysis that can make simulation models powerful for forecasting. Simulation as a forecasting method, however, is not without its drawbacks. The drawback associated with computer simulation is that, like the statistical models above, it must be manually adjusted to some degree as the real system changes. Further, simulation models are not practical as a real-time forecasting method because, in order to

generate precise confidence intervals for the output measure of interest, the model must be run for a significant amount of simulated time.

From the review of literature thus far, it is apparent that simple, static methods will not suffice to model the complexity and dynamics of short-term hospital bed availability. However, neural networks (discussed in the following section) and computer simulation (in section 3.5) are effective complementary technologies, as was shown in the work of Kilmer, Smith, and Shuman (1997), that can lead to accurate, flexible forecasts.

3.4 Neural Networks

An artificial neural network is a model that is conceptually similar to the human brain. Specifically, a neural network employs parallel distributed processing to perform computations, and it is composed of "neurons." In the brain, neurons receive inputs from the outside world or from other neurons, and they fire an output signal if the inputs are sufficient to overcome a required activation level. The neurons are connected by synapses whose strengths change dynamically; learning takes place by the reinforcement or weakening of these connections, so that signals are processed correctly. A neural network functions in much the same way: inputs to each artificial neuron are received along weighted "synapses;" the output of the neuron depends on the weighted sum of these inputs. The following section of this literature review discusses the structure and function of neural networks, which are useful for many tasks. These tasks include prediction, function approximation, speech recognition, pattern classification, and machine control. Further, forecasting applications of neural networks are discussed, and finally drawbacks to the use of neural networks are discussed that will motivate the use of a two-phase approach incorporating computer simulation.

3.41 Structure of the brain

The human brain, weighing in at a mere 3.25 pounds (Blinkov and Glezer, 1968), can outperform the most advanced computers made by humans in performing many tasks. The game of chess is just one example of man's ability to out-think machine in many arenas. Recognizing patterns (such as a particular person's speaking voice, or face), making estimations based on incomplete and flawed data, and the adaptation of a learned relationship based on changed circumstances are other examples of areas where humans excel.

The basic unit of the brain's function is the neuron. The neuron's elementary structure is as follows: it has branches (called dendrites) to receive input signals from other neurons. The junctions between neurons, where the signal propagation takes place, are called synapses. A neuron has a cell body with a specialized area where the input signals combine (called the axon hillock). If the combination of the input signals received within a small window of time is sufficiently strong, then an output signal is fired along the output branch (called the axon), where it fans out and is given as an input to other neurons.

The neurons are interconnected as a network, and as networks go the brain is extremely large: "The multiplicity of neurons and interconnections in a human brain far exceeds that of any artificial neural network. It is estimated that the brain contains on the order of $(10)^{11}$ neurons, comparable to the number of stars in our galaxy, and $(10)^{14}$ to $(10)^{16}$ synaptic interconnections among these" (Chester, 1993). Using the more conservative synaptic estimate of 100 trillion synapses, this represents on the order of 1000 average connections per neuron. Consider a hypothetical demonstration of the parallel computing power of the brain: If a single neuron processes a single signal in one millisecond (certainly not fast by standards of today's PCs), then

after that one millisecond, that same signal is passed to 1000 neurons. If we assume in this example that the brain's neurons are all connected and passing signals to each other, then that signal has reached one million neurons after only two milliseconds, illustrating the power of the brain to compute and the reason it has inspired the development of artificial neural networks.

3.42 Structure of artificial neural networks

As mentioned above, neural networks are structured as a simplified model of the brain. As with the brain, each "neuron" of a neural network has weighted inputs, a simple processing function, and outputs (shown in Figure 1). These neurons are connected into layers, which in turn form whole networks. Networks are trained to "learn" appropriate weightings, so that the weights need not be specified and can adapt continuously.

Input units to a neural network act as the artificial analog to an external stimulus to the human brain (a pin prick or muscle contraction, for example). In a mathematical sense, the inputs are those factors that are thought to have an effect on the output of the network. In the representative neuron in Figure 1, the input units are represented by x_1, x_2, x_3, x_4, and x_5.

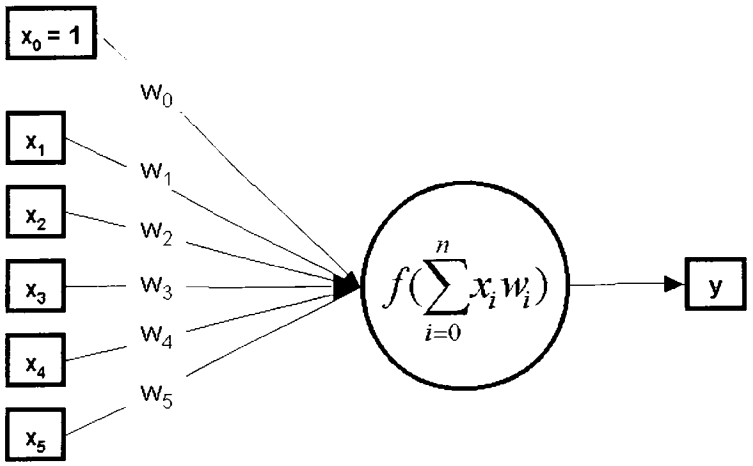

Figure 1: A representative neuron in a neural network.

The neuron's weights in Figure 1 are represented by w_0 through w5. Whereas in a biological neuron, signals may be of differing strengths, in artificial neurons both the input signal and the weight connecting it to the neuron may change. Indeed, it is through the adjustment of the weights in a network that optimal performance is achieved. This is done through training, discussed below. In this training process, the weight adjustments will be based on the errors committed by the network. To begin, weights are initialized, usually either to zero or to small random values.

The weight w_0 in the Figure above and the node x_0 are representative of the use of a bias node in the neuron. In similar fashion to the y-intercept of a linear equation, a bias allows an additional degree of flexibility in modeling, so that the relationship leamed by the network is not restrained to pass through an origin if all the input values are zero. The bias "input" is usually +1, with the weight w_0 learning in similar fashion to the other neurons in the network.

An activation function (or "transfer" function or "squashing" function) is the action taken inside the "body" of the artificial neuron to determine the neural output. The input to this function is some combination of the weighted input signals, usually as a straightforward weighted sum. This yields, as a general activation, $Y_j = f(\Sigma(x_j*w_{ji})+b)$. Here, Y_j is the neuron's output, $\Sigma(x_j*w_{ji})$ is the weighted sum of the inputs and their weights, and b is the bias value.

While any number of different effects can be achieved through the use of activation functions, three are most common. The value may be simply passed on as is (with no

transformation), it may be limited to certain discrete values (as with a step function), or it may be squashed to some value in the range [0,1] or [-1,1] (with the use of a sigmoid function). Sigmoid functions have the desirable property (seen in Figure 2 (b) and (c) below) of asymptotically approaching a desired limit as the input's magnitude increases to infinity, while becoming nearly linear when the input is between 0 and 1 (for the log sigmoid function in (b)) or –1 and 1 (for the hyperbolic tangent sigmoid function in (c)). The further importance of the types of functions shown in the Figure is that they are differentiable, which becomes important to the error back-propagation training algorithm discussed below.

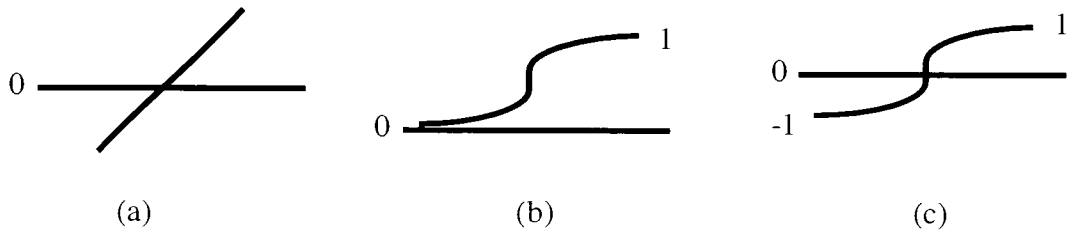

Figure 2: Neural transfer functions: (a) non-squashing linear function, (b) log sigmoid squashing function, and (c) hyperbolic tangent sigmoid squashing function.

The activity in a single artificial neuron is finished when it passes its output signal (resulting from the activation function) to all the neurons to which it is connected. At this point, the next level of network organization – the layer – comes into play. A layer of a neural network (see Figure 3 below) consists of those neurons processing the same set of input signals at the same time. A single brain neuron may be connected to hundreds of other neurons in all directions; in contrast, the most common neural network structure, called the feed-forward architecture, only has weights going from one layer to the next. That is, neurons aren't connected to themselves or other neurons in the same layer, and their outputs are not fed backward in the network. There are, however, infinite ways of arranging neural networks, and feedback and intra-layer connections are useful concepts in many of those arrangements.

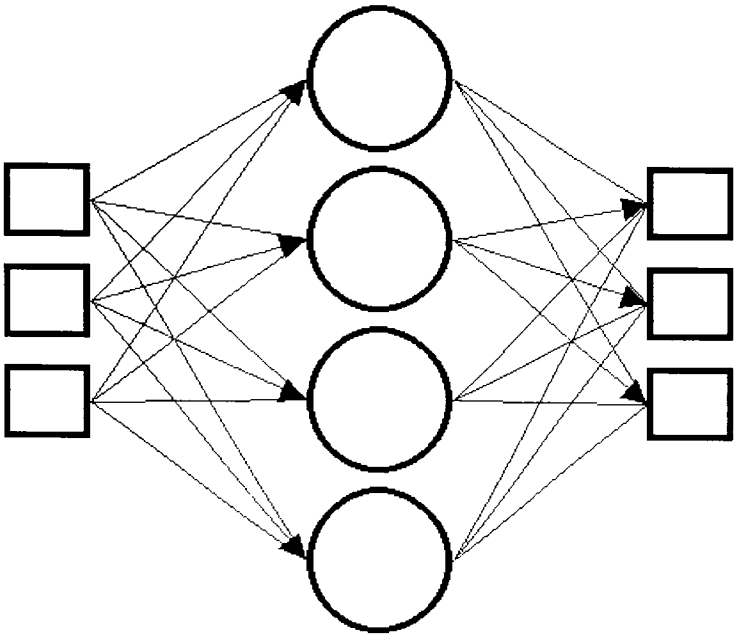

Figure 3: A representative layer in a neural network. Outputs from previous layer are represented by boxes at left; outputs from current layer are represented by boxes at right.

A neural network, then, is a collection of neurons – from one node up to many nodes in many layers – arranged as described above. The layers between the inputs and the output layer, called hidden layers, are what enable neural networks to model such a variety of different systems with accuracy. In fact, given the correct architecture for a neural network and the proper training, it can be shown to approximate any function to any desired degree of accuracy.

3.43 Training neural networks

The usefulness of neural networks is dependent, in the first place, on choosing the correct network architecture and inputs. Once these decisions are made, however, it is training that makes networks useful and enables them to be adaptive as conditions change in the system they are built to model.

With respect to neural networks, training simply means the adjustment of the weights in the network, with the goal of reducing the total prediction error the network commits. That is, for any configuration of weights, and given an input vector to the network, the output result will be different and will be closer to, or farther from, the target value depending on that configuration. While the focus of much neural network research has focused on different methods of training neural networks, error back-propagation (BP) is the method that has become most common since its introduction nearly 20 years ago by Rumelhart, Hinton, and Williams (1986). This method minimizes the total error of the network across a training data set, by adjusting the setting of each network weight proportionally to its contribution to the error. First, an input pattern is fed the network in order to generate a network output. Second, the error is calculated using a known desired output. Then, this error is propagated backwards through the network, beginning with the output neuron and working backwards. At each step, the weights are changed by an amount that depends on the input to that neuron and the neuron's learning rate. The learning rate is a parameter than can be adjusted to control the speed of training.

The number of training input vectors that are needed for successful training, as well as the number of times the set of vectors will need to be seen by the network, will vary largely with the structure of the network and the difficulty of the problem. The number of training input patterns can be roughly determined using a rule of thumb from Swingler (1996). The author's guideline is to choose N, the number of input patterns, as $N \geq W/\varepsilon$, where W is the total number of network weights, and ε is the desired level of error to reach. For a network with 25 weights, for example, where an error of 0.05 is the target, at least 500 input patterns should be used.

Considering that such a network would be small compared to practically applied networks, this guideline illustrates the need of neural networks for extensive data.

After obtaining the appropriate amount of training data, the next step must be to pre-process those data for best use in the network. The most common pre-processing involves removing outliers and scaling the data. Scaling is done so that the network's various inputs do not individually influence the network output too much, and the most common scaling is into an interval similar to (or the same as) the output interval that is desired.

A network will see each input pattern numerous times in order to leam the appropriate weight settings to minimize error over the whole data set. Each instance of presenting all the input pattems is one pass or epoch. The number of training passes, in conjunction with the size of the network and the numbers of training patterns in each pass, must be controlled to ensure the best possible training. If too many passes are performed, the network will eventually fit every data point of that set closely, at the sacrifice of not being able to generalize to new information in the future. However, too few passes (or too little data or too small a network) will result in insufficient training to model the training data or any future data. Commonly, networks will be trained to reach a certain performance goal, so that (for example) the average error or the maximum error is below a certain threshold.

Lastly, once a network has been sufficiently trained, its further usage may be in one of two modes that will have a significant impact on its adaptability. In essence, the modeler must choose whether the network will continue to learn from new data or simply use its current

learning going forward. If the network is not set up to incorporate some type of feedback, then a network in simulation mode is a static network – that is, no learning will occur. This is a further consideration for practitioners when establishing networks that will be intended to work over time in the real world.

3.44 Neural networks applications in forecasting

An important application of neural networks is in forecasting. Because neural networks can, when properly built and trained, mimic virtually any input-output relationship, they are well suited to this task of predicting a system's future performance. Further, the ability of these networks to continue fine-tuning their learning as new data are assimilated is a key asset in environments that are complex and dynamic. Below, two forecasting applications of neural networks are discussed as illustrations of the benefits of using neural networks for these tasks.

A first example is provided by Hsu and Yang (1991), who use neural networks for short-terrn electric load forecasting. This problem addressed in this paper – the short-term forecasting of a resource utilization measure – bears important similarities to the current thesis problem, and the authors illustrate that these types of problems can be successfully addressed using neural networks. The utilization rneasures in question, peak load and valley load, are each forecasted by a separate network in the authors' work. Each network's inputs are temperature data from the current time, as well as temperature data and electric load information from past observations. Further, the days are split into types of days: "Sundays and holidays; Mondays and the day after a holiday; Saturdays; and normal days" (p. 416). Thus, the time-dependent trends in this system are at least partially accounted for in the design of the neural network and selection of its inputs. The resulting network for peak electric load forecasting commits errors of only around 1% or

less for each of the 24 hours in the prediction horizon, with average error around 0.5%. By incorporating past information and explanatory variables into a neural network structure, the authors are able to achieve very close performance on this problem and declare "accurate forecasting of hourly loads can be achieved by the neural network in a very efficient manner" (p. 418).

Walczak, Pofahl, and Scorpio (2003) provide a second key example of a neural network application in forecasting, by determining emergency patient needs based on "information that is available within the first 10 min (without the use of invasive tests) of the patient's arrival" (p. 446). Here again, two separate networks are used. However, in this work, the two networks address two different types of patients (pediatric trauma patients and pancreatitis patients) that require qualitatively different types of care, rather than to model two different measures of interest. The authors use a single network (for each patient type) to predict both the severity of a patient and that patient's expected length of stay (LOS): "The likely correlation between injury severity and LOS indicates that if characteristic variables of injury severity can be identified, they can also be used to model LOS" (p. 446). The inputs for each model include information like type of injury, blood pressure, and other information readily available to the potential users of such a system. The resulting pediatric trauma LOS prediction model is able to predict LOS within one day for 52% of patients, in a population where LOS ranged from zero to 146 days. Improvement in predicting the LOS allows better planning in the hospital, and it also has an important effect on bed utilization (with higher utilization resulting from more longer-staying patients, for example). Clearly, neural networks are a capable technology for filling the forecasting needs of health care systems.

Drawbacks of neural networks

While neural networks have been shown to be excellent tools for short-term utilization forecasting, there are important limitations in their ease of implementation. First, neural networks require an extensive amount of data for training. This is because of the way that networks are trained – the weights are changed by (usually) small amounts with each presentation of inputs. As referenced above, Swingler (1996) recommends use of a guideline for determining training data needs: based on an acceptable level of error, ε, and the number of weights in the network, W, the number of training patterns should be $N \geq W/\varepsilon$. That is, if the acceptable error is 0.05, then at least 20 input patterns should be available for every weight in the network. Put simply, as the network grows in size, the number of input pattems needed for training increases rapidly.

In addition to the data needs for training neural networks, a second pitfall is in the selection of network inputs and architecture. An element of trial-and-error is apparent in the vast array of neural architectures that have been chosen to model different problems, and even to model very similar problems. Computer simulation, as used in Kilmer, Smith, Shuman (1997), offers a convenient way to address the concerns of training data and input selection. As mentioned previously, simulation models capture the richness and randomness of processes in detail and allow a treatment of whole systems. The next section explores computer simulation and highlights its usefulness as a complement to neural networks for forecasting.

3.5 Computer Simulation

3.51 Definition of computer simulation

Computer simulation is a tool used to analyze and optimize real-world systems. In contrast to statistical models, which model a particular measure from a system, computer

simulation models provide a simplified process model of the whole real-world system. From these models, analysis can be made on a host of measures (including examples like resource utilization, entities' time in the system, and lengths of queues that build in front of various resources).

Entities are the objects (or people) that flow through the process and are acted upon. In terms of health care, patients are the primary entities in any simulation model. Isken, Ward, and McKee (1999) and Cahill and Render (1999) both corroborate this choice, and indeed, the patients are (for most hospital modeling situations) the main people or objects of interest moving through the system. Entities in simulation models are assigned various attributes, which are used to create various behaviors. Using the example of hospital patients, attributes can specify a patient's diagnosis, the length of time a certain type of patient remains in a given hospital unit, the severity of the diagnosis, or any other patient characteristics. Further, the arrivals of entities to the system can be controlled to any volume or degree of randomness desired, with statistical distributions used to generate random arrivals. Arrivals can also be modeled on schedules, so that (as in hospitals) we may show more elective patients arriving in the daytime during the week and more emergency patients arriving at night and on the weekends. In short, the modeler controls the degree to which the model reflects the actual system.

The machines, people, or other things acting on entities are modeled as resources in simulation models. In many models, resources are fixed as to their location; however, there may also be types of moving resources (such as transporters moving parts from station to station). Importantly to the present thesis, hospital beds are commonly modeled in the literature as

resources. Isken, Ward, and McKee (1999), Cahill and Render (1999), Harper and Shahani (2002), and others employ this convention in their respective works. Doctors, nurses, and other hospital staff would also commonly be modeled as resources in hospital simulations. The utilization of these resources is then easily tracked through the simulation software, so that its results can be useful for analysis. As with entity attributes and arrivals, resource characteristics can be controlled to many different levels of desired control, allowing the modeling of resource efficiencies, down times (or work breaks), and schedules of resource capacities. Furthermore, resource queues are an essential part of simulation models that help determine the performance of the system and the length of time entities will wait to be served by a resource. The combination of entities, resources, and queues in a simulation model allows a wealth of different systems to be modeled to virtually any desired degree of accuracy.

3.52 Applications of computer simulation in health care

The applications of such a rich technology as simulation are, of course, many and extremely varied. In general, however, one may say that within the context of system analysis, and there are two main categories: optimization (asking the question, "How can the system be improved?") and system design (asking "How would a given system operate if it were built?"). Design models allow the modeler to see systems that do not yet exist, while optimization models focus on the current system and changing variables within it. The common theme, again, is that the analysis of these systems (whether they exist yet or not) may be conducted at a substantially lower cost and risk than implementing the same changes in the actual system (or building the proposed system without modeling).

A first example of simulation modeling for optimization is the work of Harper and Shahani (2002). The authors build a comprehensive, whole-hospital model and apply the model to study bed capacity planning (at a monthly level). Their model uses patients as entities moving through the hospital from when they enter until they are discharged. Arrival distributions and length of stay (LOS) distributions are used to model the randomness of the hospital patient processes, and pools of beds are modeled as resources that the patients attempt to use upon entering the system. This model includes logic to remove patients from the system if a bed cannot be seized in a reasonable time, mimicking the actual practice in full hospitals of transferring emergency patients and re-scheduling elective patients.

The authors' argument for the use of in-depth simulation analysis of health care systems results from the assertion that "capacity planning based on a commonly adopted simple deterministic approach will usually result in misleading results, often underestimating hospital requirements" (p. 11). The approach being criticized is the use of simple averages of patient LOS to calculate bed needs, failing to consider seasonal variations. In one case study, in contrast, their simulation model was within 5 beds of the actual occupancy for every month (versus a maximum difference of more than 30 beds for the simple-average method). Using the model, the authors are then able to recommend an improved bed allocation policy, recognizing seasonal variations and the effect of allocation on both utilization rates and patient refusal rates.

A second case study in this paper is the application of their model to optimize bed allocation for a specific patient type – respiratory patients – and to quantify the effect of bed planning for this group of patients on the entire system. "Before the model was used, managers

had provisionally decided on a capacity of 25 beds kept constant across the year, based primarily on rough deterministic calculations. Their opinion was greatly changed after the modeling study... [The hospital] revised its figures to more closely match the seasonal demand for beds" (Harper and Shahani, 2002, p.16). This example of modeling for optimization illustrates the power of an accurate simulation to help determine performance-improving policies, and it shows the flexibility of modeling, in which a single model can be re-used to study many different problems.

Blake, Carter, and Richardson (1996) offer a second example of optimization-motivated modeling in their study of factors influencing Emergency Department (ED) waiting times. After studying the very complex and subtle ED process going on in a large children's hospital, the authors apply a model to determine the factors with most influence on waiting time and to seek to reduce that time. This model is different in that it models doctors and other human resources, whereas the whole-hospital model discussed above only incorporates beds as resources. The authors are able to demonstrate in this case that by increasing physician coverage in the ED strategically, and by diverting less urgent patients to a "fast track" facility, "the proportion of patients waiting more than two hours before being seen by a doctor could be reduced by as much as 37%" (Blake, Carter, and Richardson, 1996, p. 271). The flexible nature of simulation, and performance improvements of this nature, are two compelling reasons for the increasing use of computer simulation in addressing health care problems.

A simulation-for-design example is in the work of Cahill and Render (1999), who model a hospital intensive care unit (ICU) for scenarios before and after a planned increase in ICU

telemetry beds. By incorporating three units – an ICU, a general medical floor, and a telemetry unit – into their model, the authors are able to examine effects both in the ICU and elsewhere in the hospital system. Indeed, the results of this simulation study allowed the authors to conclude that the planned ICU bed increases would improve LOS and utilization in the ICU, but that, "unexpectedly, increased ICU bed availability resulted in increased telemetry and medical floor bed utilization downstream and increased length of stay on the medical service as the proportion of post-ICU patients increased on the floors" (p. 1573). The modeling of whole systems and the analysis of effects on inter-related pieces of those systems, which is critical in modeling health care situations, is a key benefit of computer simulation over other methods of analysis for these problems.

3.53 Drawbacks of computer simulation; Neural networks as a complementary technique

While computer simulation is ideal for modeling and analyzing complex systems, the tradeoff is that the simulation becomes unwieldy for everyday use in an environment where predictions are being made based on simulation results. The awkwardness of simulation in complex problems comes from the amount of time it takes to ensure a precise response through construction of some type of statistical confidence intervals of the parameters in question.

The important drawback of needing to create confidence intervals on simulation performance measure estimates is that the total simulation time required to generate an accurate result becomes impractically long, in the case of a complex simulated system. Consider, as an illustration, a case where a single simulation replication may take less than one minute. 500 or more replications may be needed to achieve the required accuracy in the estimate, and this will

greatly increase the amount of time it takes to create an estimate. While producing accurate results, simulations may take too long to do so.

The second important drawback to the use of computer simulation models is in their static nature: simulation models are typically a picture of a current system and so do not adjust as patient case mixes, numbers of beds in units, and other variables change. This requires the periodic intervention of a simulation expert to update the model, so that it may remain useful to its users. Other modeling techniques will also only be valid for a system within certain ranges of change, however, the point bears mentioning for a technique into which a large portion of time may be devoted in model development, and for which a typical hospital has few or no experts to make needed changes.

The strengths of computer simulation and neural networks may be synergized into a two-phase approach, like the approach cited in Kilmer, Smith, and Shuman (1997). Again, in this work, the simulation model is used to gain an understanding of the system being modeled, and from this understanding, the important input variables for a neural network may be identified. Further, by creating and using a valid simulation model, the volumes of needed training data may be easily created for use in the network. In turn, the neural network overcomes the major drawbacks of computer simulation: its slowness for creating real-time forecasts with confidence intervals, and its static nature. Once the network is trained, it can generate forecasts in a matter of seconds, and if properly built, the network can continue to learn as new data are available. Thus, the literature shows that methods such as computer simulation and artificial neural

networks are well suited to the task of short-term hospital bed availability forecasting, and that the most benefit may be derived by incorporating each technology to complement the other.

4. Methods

The methods in this section were developed as a general method for forecasting short-term bed availability. Generous assistance with information on system layouts, and aggregate monthly and yearly data that were critical to the success of this project, were provided by the Park Ridge Hospital in Rochester, New York. Park Ridge is an average-sized hospital of around 200 beds, offering a range of services. The generic model developed in the following sections is based loosely on Park Ridge Hospital and on the data they provided.

4.1 System Description

The generic hospital conceived for modeling in this research is a 200-bed hospital providing an average range of services. Patients are treated in a succession of units within the hospital and then discharged back into the outside world.

4.11 Hospital size

The number of beds in a hospital is a common measure of size. According to the Agency for Healthcare Research and Quality (2004), 190 beds approximately an average size for a hospital, and 200 was chosen as the level for this system. These 200 beds are comprised only of the sum of medical floor beds, intensive care beds, rehabilitation unit beds, and chemical dependency unit beds. Labor decks, surgical suites, post-operation beds, ED gumeys, and bassinets are excluded from the count. The reason for these exclusions is that when patients are in one of the excluded beds, they are also occupying another bed (except in the case of gurneys and bassinets). That is, when a patient is in surgery, for example, that patient also has claim to the medical bed he/she will occupy following the surgical process. Thus, inclusion of these beds would lead to double counting of many patients in the system. ED gurneys are excluded since emergency patients are not considered to be admitted to the hospital. If a patient stays past 23

hours or needs to be admitted for another reason, then the patient will be officially admitted into a different type of bed. Bassinets are excluded from the number of hospital beds because the arrival of newborns is directly dependent on the arrival of mother patients, and the utilization of bassinets is therefore dependent on the utilization of medical floor beds for mothers.

4.12 Hospital layout and patient flow

The generic hospital has units for admissions, emergency, surgery, post-operation observation, intensive care, labor, nursery, medical care, chemical dependency, and physical rehabilitation. Based on conversations with experts, this layout captures an average picture of hospital operations at a broad level of detail. At this level of detail, many hospital departments (for example, x-ray areas and laboratories) do not affect bed utilization directly and so are excluded. Further, while other hospitals may have additional units or not have some of the units indicated for this hospital, this selection is a good representative system for purposes of modeling and for measuring bed utilization. The layout of the units in the generic hospital model is illustrated below in Figure 4.

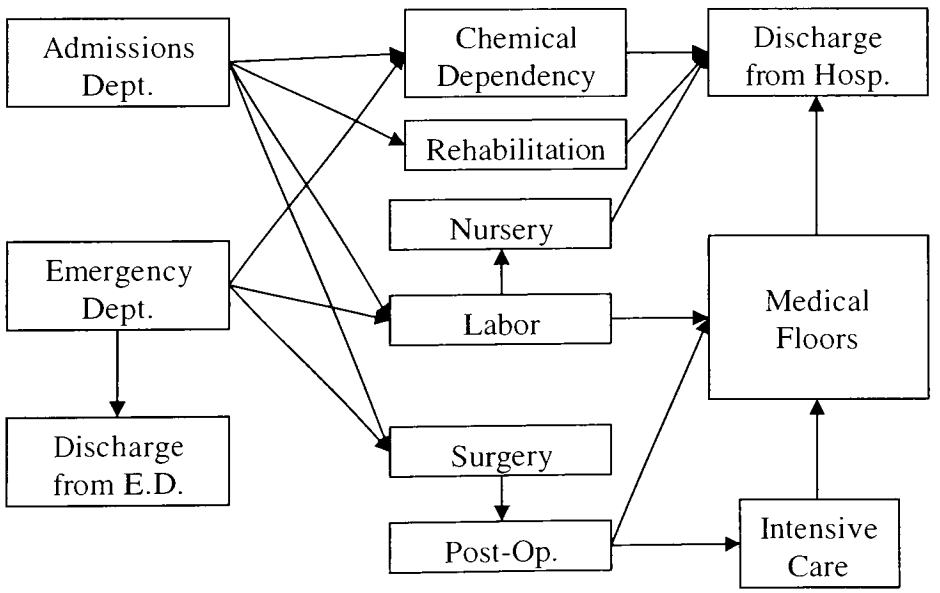

Figure 4: Flow diagram showing the flow of patients through the generic hospital.

The arrows in the above Figure indicate the paths along which patients can flow, depending on their diagnoses. Patients arrive either through the Emergency Department or through Admissions. In the ED, patients are triaged upon entry and assigned a diagnosis and a severity. Patients then wait to occupy a gurney, where they are treated until they either exit the hospital from the ED or are admitted into the hospital. Some patients (pregnancies and heart failure patients) will bypass the gurney process and go directly to surgery or labor, respectively. The remaining patients that are admitted from the ED will go to surgery, a medical floor, or the chemical dependency unit. In Admissions, no triage is performed, since patient diagnosis and severity are known on arrival for non-emergency patients. Patients wait to be assigned to a bed; then, they will go to surgery, labor, chemical dependency, or rehabilitation.

In each unit of the generic hospital, a patient's unit-level length of stay depends on his diagnosis. After receiving care in a given unit, the patient proceeds to the next service needed, until he is ready to be discharged from the hospital. The paths followed by different types of patients through the hospital are shown in Appendix I. If the patient is ready for discharge during the day (between 8:00 am and 8:00 pm), he will leave the system, and the bed that was occupied by the patient is cleaned and prepared for the next patient that will occupy it. The time required to clean and prepare a bed is approximately 30 minutes. If the patient is ready to leave during the night, he will be delayed until the following day. This discharge practice is common among hospitals, where the doctors who approve patient discharges work only during the day.

4.13 Patient arrivals and volume

The generic hospital used for modeling in this research is like real-world hospitals in that its patients arrive somewhat randomly according to some general cycles. There are two streams

of arrivals in the generic hospital: one stream enters the Admissions department, and the other stream enters the Emergency Department. In terms of daily trends, non-emergency patients tend to arrive in a cluster in the early morning since their procedures are scheduled during the day. In contrast, emergency patients arrive more randomly, usually during the evening hours (after work hours or late at night). Beyond the daily trends, though, there are also week-level trends for the arrivals of these patients. Scheduled patients will tend to arrive early in the week, so that these patients can recuperate from their procedures and be discharged before the weekend. The ED patients, however, are more apt to arrive on the weekends as a result of incidents like weekend sports injuries and motor vehicle accidents.

The volume of arrivals to the generic hospital is based on an average weekly number of arrivals to the case study hospital, since the case study hospital is also of an average size. This total number of arrivals per week is then split among the two arrival streams, with one-third of patients arriving as emergency patients. This proportion is consistent with information from the Agency for Healthcare Research and Quality (1997). A further breakdown of patient volume is the number of patients arriving in each arrival stream with each diagnosis. These percentages within arrival streams are determined by using data from the AHRQ (2004) Health Care Utilization Project.

4.2 Computer Simulation Model of the Generic Hospital

Phase one of the general forecasting method outlined in this research was the construction of a computer simulation model of the generic hospital system. The building of the model followed the generic system as closely as possible; this section details the construction, verification, and validation of that model.

4.21 Modeling software

The simulation software used for this thesis is Arena © 7.01. This software allows great flexibility in modeling, and it also allows construction of convenient animation for troubleshooting and communication purposes. Arena has been used often in modeling health care systems. For example, Cahill and Render (1999) have used Arena to gauge ICU bed utilization and determine the merits of an expansion plan.

4.22 Patient types

Patients were modeled as entities, as is the standard practice for hospital simulations. However, it is important to reflect the differences between different types of patients with respect to their path through their hospital and length of stay, rather than using a single patient type. For example, deliveries are will use the labor and medical floors before exiting, and their median length of stay is around two days. In contrast, rehabilitation patients are admitted directly to (and leave from) the rehab floor, and their expected length of stay is well over a week. For this model, twelve representative types have been chosen to model a variety of patient flows and lengths of stay. The patient types that are included in the generic hospital model are shown in Appendix I, along with the flow and average length of stay of each type. Once representative patient types were chosen, national statistics on admissions by patient type (from the AHRQ (2004) Health Care Utilization Project) were used to calculate relative frequencies of these patient types in the model.

The attributes given to the patients were patient type, severity, bed type, sequence number, and unit length of stay (LOS). Patient type is assigned upon the patient's entry into the hospital (either through Admissions or the ED), based on their relative frequencies as mentioned above. Severity is also assigned according to a discrete distribution, as a way to prioritize the

typically more urgent emergency patients in the usage of surgery suites, labor decks, and other beds. Bed type is assigned when a bed is assigned to a particular patient, so that the patient may move through the hospital and retain that bed. Sequences were designated to stipulate a different path of care for different types of patients. The ability to show the differences between different patient types is clearly important in this type of modeling, and within each patient type's sequence, the unit LOS is assigned as patients move between units.

4.23 Patient arrivals

The arrivals of patients were modeled by two arrival schedules in the simulation: one schedule is for elective (non-emergency) patients, while the other is for emergency patients. Two streams of arrivals were chosen because emergency and elective patients have different arrival trends. These arrival trends of each stream of patients and its breakdown throughout each day of the week were determined based on discussions with experts; actual hourly arrival data were not available for empirical trend determination. Figure 5 shows a graph of the arrival schedules for elective and emergency patients. Note in the Figure, that elective patients (in panel (b)) comprise a larger number of patients than emergency patients, shown by the different vertical scales in each panel. Percentages of arrivals of each patient type were determined using the relative frequency of each patient type, discussed above. These percentages were then manipulated to account for all the patient types not used in the model, so that the mix of patient arrivals is representative of what a typical hospital might experience.

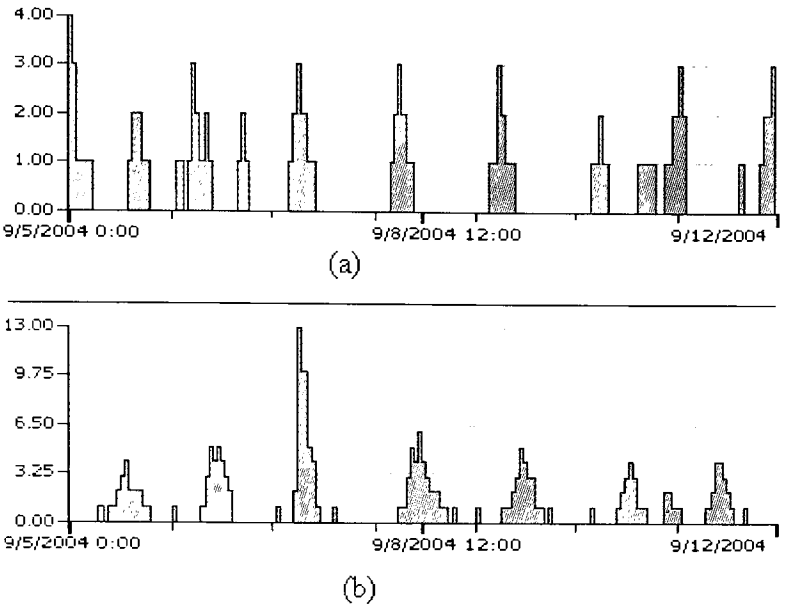

Figure 5: (a) Emergency arrivals and (b) non-emergency arrivals to the generic hospital model. Note the difference in scale (maximum emergency arrivals in one hour is 4, versus 13 for scheduled patients).

4.24 Resources

The sole resources modeled in the simulation are beds; the modeling of beds as resources is consistent with common practice for hospital simulation models. The beds were allocated to each unit, so that unit-level views of bed availability are also possible and the number of patients on each unit can be realistically modeled. Each pool of beds has an associated queue to "hold" waiting patients for each bed type. The queue discipline for these bed queues is to choose the highest-priority patient first (using the severity attribute), and within the same severity level to employ first-in-first-out logic. The flow of patients through the hospital model is govemed by the generic hospital setup (see Figure 4) and the pathways identified in Appendix I. At each unit, patient length of stay is patient type-specific, according to a triangular distribution assigned in that patient's sequence. The discharge logic from the generic system is then applied to determine when patients exit the hospital.

4.25 Model outputs

The outputs of interest from the model were the hourly counts of each type of patient and the count of available beds each hour. Output files were set up to capture this information, facilitating its use in the following phase of the general forecasting method. Further, Arena enables a large number of outputs to be automatically captured; in this way, the utilization levels of each bed type, the length of queues for each bed type, and average length of stay (averaged over all patients) were automatically tracked as well.

4.26 Verification and validation

Model verification was done to ensure that the simulation model functions as intended. To verify, the simulation was first de-bugged and then run to look for abnormal performance. Utilization plots for the various medical floor bed pools helped show how the various medical floors filled up through the simulation run. Verification was key in capturing several errors that, once corrected, allowed the model to run smoothly.

Once the model was functioning, a second verification procedure consisted of changing variables in the model to observe whether the model would still run correctly. The following scenarios were run, each altering one variable in the model: constant arrival rate instead of scheduled (time-varying) arrivals; decreased "scale" of the system, with significantly reduced arrival volume and number of beds; increased scale, with significantly increased patient volume number of beds; and finally a scenario with equal percentages of all patient types within each stream in the patient type distribution. These scenarios are summarized in Appendix III. The verification scenarios all ran successfully, with the caveat that small adjustments had to be made to maintain reasonable conditions in the model. That is, for example, if the number of patients is increased without increasing the number of beds, the number of patients in the system waiting

for beds increases without limit. On a more subtle scale, the number of beds dedicated to rehabilitation patients had to be increased under the equal patient mix scenario, as these patients have a far longer length of stay than the other types of patients in the model (11 days, as opposed an average of under 4 days for the other patient types).

Model validation is usually accomplished through comparison of a simulation model with the real system it is modeling, both in terms of appearance and output results. As the model was built to match the generic system, the similarity of appearance is assured. With respect to model performance, a number of comparisons between model output and expected outputs were performed. Comparisons based on number of patients entering the hospital, average LOS for all patients, and patient case mix were performed and are summarized below in Table 1.

Table 1: Validation results for generic hospital simulation model.

Measure	Expected Value	Actual Value	% Difference
Number of patients entering the hospital (per day)	36.71	34.48	6.07%
Average LOS (all patients)	173.99	168.24	3.30%
Percent Deliveries	25.00%	24.71%	0.28%
Percent Chem. Dep.	15.00%	14.87%	0.13%
Percent Fractures	10.00%	9.69%	0.31%
Percent Hip Replacements	10.09%	10.33%	-0.24%
Percent Rehabilitation	6.73%	7.20%	-0.46%
Percent Medical Back Issues	16.82%	17.58%	-0.75%
Percent Chest Pains	8.18%	7.47%	0.70%
Percent Heart Failures	3.27%	3.32%	-0.04%
Percent Hemorrhages	4.91%	4.83%	0.08%

The results of the validation show that the simulation model has been able to reproduce the general conditions of a representative hospital. Further performance testing of this simulation model is not possible without real data against which to compare the model's outputs.

4.27 Experiment to determine the effect of patient mix

Patient mix was theorized to have a significant effect on bed availability. Confirming or refuting this effect was important in deciding whether to include this information as an input to the neural network in phase 2. Therefore, an experiment was undertaken to verify this effect. An altemative simulation model was built using a different mix of patients: all patient types were modeled as equally likely within each arrival stream. The weighted average LOS for the base model, calculated as the average LOS of each patient type times that type's percent of the population, was around 94 hours. The weighted average LOS for the alternative model was 111 hours.

Once this model had been built, keeping all other details the same as the base model. 100 weeks of bed availability data were captured to conduct a 2-sample T test for equal means. The resulting T test, and a boxplot comparing the bed availability (expressed as a percentage) of the two models, is given in Figure 6 below ("sm4_1" indicates the base model and "sm4_5" indicates the altemative model). Clearly, an equal patient mix was shown to significantly decrease hospital bed availability versus the base mix (the estimated difference in availability being 0.073, or roughly 14 beds).

```
Two-sample T for sm4_1 vs sm4_5
           N    Mean   StDev  SE Mean
sm4_1  16800  0.2513  0.0773  0.00060
sm4_5  16800  0.1783  0.0677  0.00052

Difference = mu (sm4_1)   mu (sm4_5)
Estimate for difference:  0.072999
95% CI for difference:  (0.071445, 0.074553)
T-Test of difference = 0 (vs not =): T-Value = 92.09
P-Value = 0.000  DF = 33598
Both use Pooled StDev = 0.0727
```

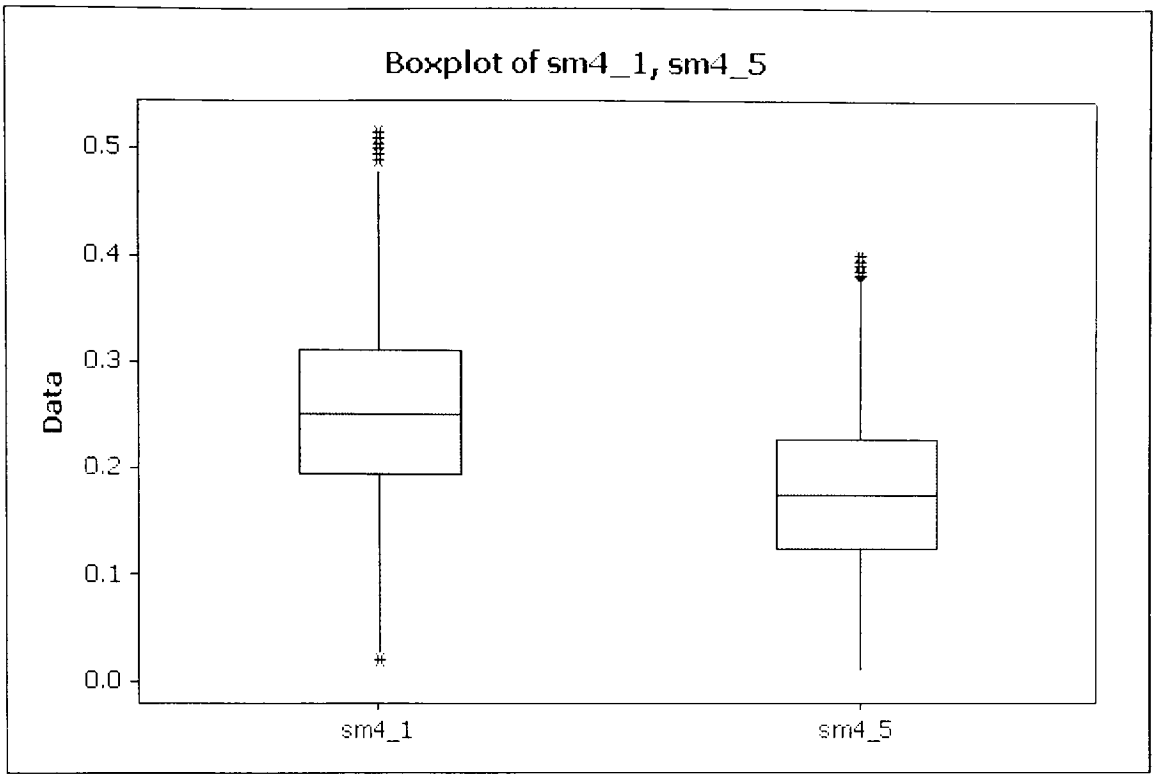

Figure 6: comparison of means for base model (here "sm4_1") and alternative model.

Based on the result of the above T test, patient mix was concluded to be a significant effect in determining hospital bed availability, and it was included in the neural network, which is described in the following section.

4.3 Neural Network for Bed Availability Forecasting

Phase two of the general method developed in this thesis is the creation of a neural network forecasting system, relying on the previously discussed computer simulation model for training data and assistance in choosing input variables. This section discusses a network to forecast, one time step into the future, the expected value of the number of available beds in the hospital. A tolerance interval is created from the mean squared error of a test data set and can be updated as the network is used. Investigation of a second network to estimate the forecast error of the first neural network is also discussed.

4.31 Network inputs

The inputs to the expected value network (EVN) are the current hour of the week, the current number of beds available, and the current count of each type of patient in the hospital.

The use of time as an explicit input is a novel choice in this thesis research. While other papers reviewed accounted for time dependency and seasonality, this was done primarily through the addition of model terms for seasonal effects (as in Jones, Joy, and Pearson, 2002) or through de-seasonalizing the data. This network input structure uses an indicator node for each hour of the week, resulting in a 168-element vector for this input. By choosing this structure, the neurons for each hour of the week have separate weights to adjust, allowing the different availability trends to be modeled. An alternative method of explicitly incorporating time as an input would have been to use the magnitude of the current time (i.e., "150" for hour 150 of the week, "1" for hour 1, and so on), requiring a single input node and single input weight set. However, Figure 7, which shows the availability distribution at each hour of the week over 300 weeks, highlights the different trends between hours and the resulting importance of separate input weight sets for each hour, rather than one shared set of weights.

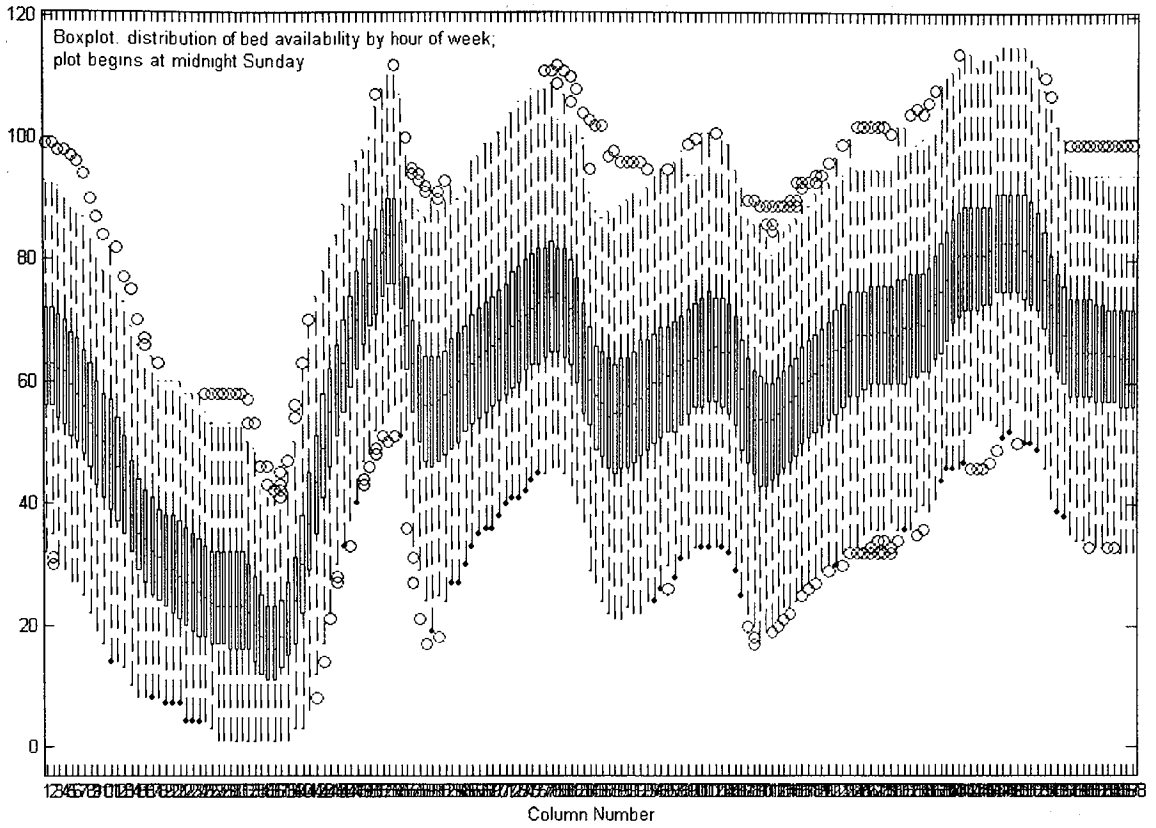

Figure 7: Boxplots of bed availability for each hour of the week, over 300 weeks.

The second input is the count of patients of each type in the hospital at the current time. In the experiment described in the previous section, the effect of this input variable on bed availability was confirmed: specifically, as the number of longer-staying patients in the system increases, the availability of beds decreases. The hourly count of each type of patient was used, yielding a 12-element vector representing this input. The current number of available beds, used as the third input to the network in the generic hospital forecasting system, is a good predictor of the number of available beds one hour ahead, because of the small change from hour to hour. However, this method is not limited to time steps of one hour – an arbitrary time step (30 minutes, six hours, twelve hours, etc.) may be used within this method. A verification of the effectiveness of this EVN structure is shown in a later sub-section, for a six-hour time step.

Beyond the effect of each of the above variables on bed availability, it is also of note that the chosen inputs are all readily available in hospitals. That is, it is no difficulty to observe, at any point in time, the number of each type of patient (or, for practical purposes, a selection of patient types as used here), the number of available beds, and the current time. This is a differentiating aspect of this thesis approach from other previous research cited earlier. The work of Walczak, Pofahl, and Scorpio (2002) uses similarly available inputs; however, their application is limited to a single hospital unit and was focused on different measures. The work of Kilmer, Smith, and Shuman (1997), in contrast, addresses a similar problem as the current thesis, but their choices of inputs include information that would not be readily available (and would compromise the need to use a neural network at all).

The resulting input vector consists of 181 elements to represent current hospital conditions (168 for the current time, 12 for the counts of patient types, and one for the current bed availability). The scaling and use of these inputs is described in the sub-section below on network training.

4.32 Network architecture

The goal with respect to the architecture of this network was to provide a simple, effective structure for forecasting the expected value of bed availability one step ahead. Consequently, the network was tested with a single processing layer of 168 linear neurons, also using a single non-learning output neuron to sum the results from the processing layer. Linear neurons were chosen for simplicity. In training and testing with several models, no need for non-linear transfers was found. The inputs described above were used in the architecture, and the single output value was the bed availability a single time step ahead.

The correct number of hidden-layer neurons is shown in many sources to be difficult to determine optimally, depending on a number of factors. These factors include numbers of inputs and outputs, the variability in the data, the complexity of the input-output relationship, the hidden unit activation chosen, and more ("How many hidden units", 2004). The overall goal is to provide adequate hidden units to allow the network to learn the function of interest, while avoiding over-fitting of the training data that will be used.

Initially, a large number of hidden-layer neurons (168, corresponding to one for each day of the week) were chosen. However, an experiment conducted with smaller networks (84 and 107 neurons) yielded interesting results. The experiment involved creating three networks (with 168, 84, and 107 hidden neurons, respectively) and training them on the same data set to the same number of training passes (200 was chosen). The 84-node network was chosen for the experiment because it is half the size of the original network. The 107-node network was chosen as a size in between the other two networks. The results of this training showed that the largest network created the smallest maximum training error and the smallest mean squared training error, as shown in the Table below. However, the smallest mean squared error using a novel test set was exhibited by the network with 107 neurons. Although its maximum error on this test set was higher than the others, the mean test errors illustrate that the 107-node network had its errors distributed more evenly around the zero line.

Table 2: Test results for various network sizes, for determining network hidden layer size. Results are after 200 passes of training on each network.

	168 Neurons	84 Neurons	107 Neurons
MSE (training error)	6.39 beds^2	10.67 beds^2	8.43 beds^2
Maximum Absolute training error	12.38 beds	13.67 beds	13.89 beds

MSE (test error)	463.13 beds^2	204.32 beds^2	164.92 beds^2
Max. Absolute test error	41.56 beds	38.87 beds	43.52 beds
Mean test error	20.11 beds	-12.96 beds	-6.83 beds

4.33 Network training

The training algorithm used for the EVN is error back-propagation. Its relative simplicity and widespread use were the main motivations for its adoption. Error back-propagation is discussed in more detail in section 3.4 of the literature review.

Fifty weeks of data, generated from the validated hospital simulation model, were used as the training data set for the EVN. In earlier neural network experiments, and following the rule-of-thumb of Swingler (1996), the number of desired training patterns was calculated as $N = W/\varepsilon$, where ε is the approximate expected level of error. Using $\varepsilon = 0.05$ resulted in 20 patterns for each weight in these earlier experimental networks. The choice of 50 weeks of patterns (8,400 values) was made as a compromise between this guideline (which would have resulted in a requirement of over 600,000 input patterns) and the capabilities of the machine on which these experiments were carried out.

The training data set was run through the neural network training algorithm until the maximum absolute error committed by the network was ten beds. The number of training passes required to achieve this performance is listed in Table 3 below. While the training could be continued to be arbitrarily precise, continued training can lead to over-fitting. See section 3.4 of the literature review for more discussion of the trade-off between accuracy and generalization.

Along with the training algorithm, training data, and amount of training chosen for neural network training, the network's learning rate is another important specification for successful training. As discussed in previously in the literature review, the leaming rate modulates the amount by which weights will be changed during training. A learning rate of 0.02 was used to train the EVN. This rate was chosen to be conservative: while a lower learning rate causes slower learning, it avoids the problem of network weights being changed too rapidly and so missing their optimal values.

The resulting trained neural network provided good performance with respect to the training data set, as shown in the following Table of training results for the EVN.

Table 3: Training results for the expected value network.

Maximum training error	10 beds
MSE (training error)	3.84 beds^2
Number of passes	400
Total input patterns presented	3.36 million

4.34 Building forecast tolerance intervals

It is important to provide, when forecasting, an indication of precision to the estimate being made. This was achieved in this method by the use of a tolerance interval on the expected number of available beds one time step ahead. The tolerance interval was based on the computation of the mean squared error (MSE) for the training data set, on a simulation (non-learning) run of the data through the trained network. The errors from this run were each squared, and the mean of this sum of squared errors was used in the following equation for a tolerance interval from Askin and Goldberg (2002):

$$\text{Half-width} = (\text{MSE})^{1/2} * Z(95\%) = 1.96 * (\text{MSE})^{1/2}$$

The resulting 95% tolerance interval half-width for the EVN was 3.84 beds. This half-width was then applied during the network testing to create tolerance intervals for the forecasts, as described in the next sub-section.

4.35 Testing the neural networks

In order to test the performance of the neural network, a second set of data from the hospital simulation model was obtained. This data set consisted of five weeks' worth of data for testing. For each input set, a prediction was generated from the network, and the 95% tolerance interval half-width calculated above was applied. The results of the testing are shown below in Table 4. Less than 5% of 95% tolerance intervals are missed using this forecasting method.

Table 4: test results from trained expected value network, using novel test data.

Mean test error	-0.28 beds
Maximum Absolute test error	8.29 beds
Number of tolerance intervals missed	31
Percent missed	3.7%

4.36 Neural network for predicting forecast error

In order to improve on the building and dynamic updating of tolerance intervals for this forecasting system, a second neural network has been investigated to predict the forecast error from the expected value network. This forecast error network (FEN) would allow each hour to not only have separate estimates of bed availability, but also of the error with forecasting that availability. The importance of such a difference in variability can be demonstrated by comparing the amount of arrival and discharge activity at different times of the day. During the day, patients may arrive and leave; however, at night only arrivals will be seen, and those arrivals will tend to be on a smaller scale than the arrivals during the daytime.

The inputs of this network, thus far, have been the same as the inputs for the EVN. One trial, using only the current time as an input, did not yield significantly improved results. The network's output is the absolute value of the error involved in predicting the number of available beds, and therefore it is always positive. This, in turn, has had an effect on the network architecture. The structure has been changed a number of times in attempts to increase the speed and success of training, including trying a second hidden layer (and even a third), and adjusting the numbers of neurons in these layers. The training parameters of passes, learning rate, and transfer function have also been examined. Currently, the best solution employs the same type of linear hidden layer as the EVN, followed by a log-sigmoid layer of fewer neurons to restrain the network's output to be positive.

While this network has not been successfully trained thus far, investigation into this enhancement continues and is one of the subjects for future research discussed in a later section.

4.37 Neural network for 6-hour time step

In order to illustrate the application of this method to a time step other than one hour, an experiment was performed to create an EVN for a time step of six hours.

The inputs and architecture of this network were precisely the same as those used in the one-hour network described above. The same training input patterns were also used. The sole difference, in fact, was in the use of targets six hours into the future, rather than one hour.

As the size of the time step increases from one hour to longer time steps, the error in these forecasts will naturally increase, as the current status tells us less about future times farther away. Thus, this network was not trained to the same stopping criterion as the one-hour network, as this would surely have led to excessively long training that would have over-fitted the training set. Instead, 400 training passes were used to gain perspective into the performance of this network relative to the performance of the other EVN. By coincidence, the one-hour network took 400 passes to reach its performance goal of a maximum error less than ten beds. The training performance of this six-hour network is below in Table 5.

Table 5: Training results for the six-hour expected value network, after 400 passes.

Maximum training error	28.95 beds
MSE (training error)	41.34 beds^2
Number of passes	400
Total input patterns presented	3.36 million

The training data set was simulated on the trained network, as described above, to derive the mean squared error for use with this network. The resulting tolerance interval half-width was 12.60 beds.

The six-hour network, once trained, was tested using the same test input patterns as those used above on the one-hour network. The test results are below in Table 6. Again, this performance is not nearly as good as would be expected from the one-hour network; however, this is due to the increasing distance from the current time and the practical limitations placed on the training of this network.

Table 6: Test results for the six-hour expected value network, after 400 passes.

Mean test error	-3.24 beds
Maximum Absolute test error	54.70 beds

Number of tolerance intervals missed	294
Percent missed	35%

5. Conclusions and Recommendations for Future Research

5.1 Conclusions

Computer simulation and artificial neural networks have been shown to be an excellent approach to the problem of short-term hospital bed availability forecasting. A computer simulation model has been built to model the non-stationary arrivals, various patient types, and overall complexity present in hospital systems. That model is valid for the data that was available with respect to bed utilization, patient type prevalence, and patient lengths of stay. A neural network has been created to forecast the availability of hospital beds one hour into the future, using information on the current time and current counts of patients and available beds. This network provides good performance in this task for the data sets tested. A forecasting tolerance interval has been built from the mean squared error of the network's forecasting performance, and investigation is ongoing into a network to estimate this forecast error adaptively for the creation of the best tolerance intervals possible.

5.2 Recommendations for Future Research

Many advances and enhancements to this research are possible and encouraged.

- First, as mentioned directly above, a second neural network is being investigated to estimate the forecast error of the EVN in predicting bed availability.
- A further and more fundamental advance will be to obtain more accurate, real data for comparison against the performance of both the simulation model and the neural network. In this sense, this research is the beginning of an iterative process to work toward getting the best information, to in turn create the best forecasting method.
- Another important development for this research is to create a single network to predict bed availability an arbitrary number of steps into the future (up to some limit). Research in this

area has already begun, involving the stringing together of multiple one-hour expected value forecasts.

- This method should eventually be implemented in real hospital systems. The best way for that to occur is integration into a hospital information system, so that the newest data can be used to continuously update the neural network's learning and alert when more in-depth training or refinement would be needed.
- Finally, this method can be expanded into other applications in a variety of ways – through use in other industries, through specialization to single hospital units, and particularly through expansion into a community-wide model of shared hospital capacity management. The potential impact of improved resource utilization forecasting has far-reaching implications for helping communities better utilize their shared resources.

References

Agency for Healthcare Research and Quality (1997). *Hospitalization in the United States, 1997.* Retrieved July 1, 2003 from http://www.ahcpr.gov/ data/hcup/factbk1/#hospitaladmissions

Agency for Healthcare Research and Quality (2004). *HCUPnet: Healthcare Cost and Utilization Project.* Retrieved August 10, 2004 from http://www.ahrq.gov/HCUPnet

Askin, RG and Goldberg, JB (2002). *Design and analysis of lean production systems.* New York: John Wiley & Sons, Inc.

Bagust, A, Place, M, and Posnett, JW (1999, July 17). Dynamics of bed use in accommodating emergency admissions: stochastic simulation model. *British Medical Journal, 319*(7203), 155 – 158.

Blake, JT, Carter, MW, and Richardson, S (1996, November). An analysis of emergency room wait time issues via computer simulation. *INFOR, 34*(4), 263 – 272.

Bonfà, I, Maioli, C, Sarti, F, Milandri, GL, and Dal Monte, PR (1993). HERMES: an expert system for the prognosis of hepatic diseases. *First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, 1993* (pp. 240 – 246).

Blinkov, SM and Glezer, II (1968). *The human brain in figures and tables: a quantitative handbook.* New York: Basic Books, Inc.

Bramer, MA (1982). A survey and critical review of expert systems research. *Studies in Cybernetics 1: Introductory Readings in Expert Systems,* 3 – 29.

Bureau of Labor Statistics (2004). *Employees on nonfarm payrolls by industry sector and selected industry detail.* Retrieved August 15, 2004 from http://www.bls.gov/news.release/empsit.t14.htm

Cahill, W and Render, M (1999). Dynamic simulation modeling of ICU bed availability. *Proceedings of the 1999 Winter Simulation Conference* (pp. 1573 – 1576).

Centers for Medicare and Medicaid Services (2003). *Highlights – National Health Expenditures, 2001*. Retrieved July 2, 2003 from http://cms.hhs.gov/statistics/nhe/historical/highlights.asp

Chester, M (1993). *Neural networks: a tutorial*. Englewood Cliffs, NJ: PTR Prentice-Hall, Inc.

Côté, MJ and Tucker, SL (2001, May). Four methodologies to improve healthcare demand forecasting. *Healthcare Financial Management, 55*(5), 54 – 58.

Drug-price program notes (2000, August 10). *The Wall Street Journal*, pg. A18.

Harper, PR and Shahani, AK (2002). Modeling for the planning and management of bed capacities in hospitals. *Journal of the Operational Research Society, 53*(1), 11 – 18.

How many hidden units should I use? (2004). Retrieved September 4, 2004 from http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-10.html

Hsu, YY and Yang, CC (1991, September). Design of artificial neural networks for short-term load forecasting. Part II: multilayer feedforward networks for peak load and valley load forecasting. *IEE Proceedings, Part C: Generation, Transmission, and Distribution, 138*(5), 414 – 418.

Isken, M, Ward, T, and McKee, T (1999). Simulating outpatient obstetrical clinics. *Proceedings of the 1999 Winter Simulation Conference* (pp. 1557 – 1563).

Jones, SA, Joy, MP, and Pearson, J (2002, November). Forecasting demand of emergency care. *Health Care Management Science, 5*, 297 – 305.

Joseph H. Kanter Family Foundation (2002). *The third annual conference of the Health Legacy Partnership*. Retrieved June 3, 2003 from http://www.healthlegacy.org/education

Kilmer, RA, Smith, AE, and Shuman, LJ (1997). An emergency department simulation and a neural network metamodel. *Journal of the Society for Health Systems, 5*(3), 63 – 79.

MacStravic, S (2004, August). Good old reliable (?) trend forecasting should be re-evaluated. *Health Care Strategic Management, 22,* 11 – 12.

MacNiece, EH (1961). *Production forecasting, planning, and control.* New York: John Wiley & Sons, Inc.

Rumelhart, DE, Hinton, GE, and Williams, RJ (1986, October). Learning representations by back-propagating errors. *Nature, 323*, 533 – 536.

Swingler, K (1996). *Applying neural networks: a practical guide*. San Diego: Academic Press, Inc.

Walczak, S, Pofahl, WE, and Scorpio, RJ (2003, March). A decision support tool for allocating hospital bed resources and determining required acuity of care. *Decision Support Systems, 34*, 445 – 456.

Woodruff, M (2002). *Inpatient bed need planning – back to the future?* Retrieved January 30, 2003 from http://www.bristolgroup.com/inpatient_bed_need_back_to_the_future.pdf

Appendices

Appendix I: Generic Hospital Patient Characteristics

Table I.1: Characteristics of patient types used in modeling the generic hospital system.

Patient Type	Median LOS (days)	Path Through Hospital
Delivery (non-emergency)	2	Admissions, Labor, Medical Floor, Exit
Delivery (emergency	2	E.D., Labor, Medical Floor, Exit
Newborn	1	Labor, Nursery, Exit
Chemical Dependency (non-emergency)	4	Admissions, Chemical Dependency, Exit
Chemical Dependency (non-emergency)	4	E.D., Chemical Dependency, Exit
Fracture (non-emergency)	5	Admissions, Surgical Process, Post-Op, Medical Floor, Exit
Fracture (emergency)	5	E.D., Surgical Process, Post-Op, Medical Floor, Exit
Rehabilitation	11	Admissions, Rehab Floor, Exit
Medical Back Problems	4	Admissions, Surgical Process, Post-Op, Medical Floor, Exit
Chest Pains	1	E.D., Medical Floor
Heart Failure	5	E.D., Surgical Process, Post-Op, ICU, Medical Floor, Exit
G.I. Hemorrhage	4	E.D., Surgical Process, Post-Op, ICU, Medical Floor, Exit

Appendix II: Distribution of Beds in Generic Hospital Simulation Model

Table II.1: Distribution of beds in generic hospital system.

Unit	Number of Beds
Medical Floor 1	20
Medical Floor 2	20
Medical Floor 3	20

Medical Floor 4	20
Medical Floor 5	20
Medical Floor 6	20
Chemical Dependency	20
Rehabilitation	20
Labor Decks	5
ED Gurneys	4
Intensive Care	10
Post-Operation	6
Bassinets	25
Surgical Suites	8

Appendix III: Generic Hospital Model Verification Scenarios

Table III.1: Hospital model verification scenarios.

		Hospital Scale		
Scenario Label	**Arrival Variability**	**Arrival Volume**	**Number of Beds**	**Patient Mix**
Base Model	Schedule	Base (257/week)	200	Base
Constant Arrivals	Constant	Base	200	Base
Srnall Hospital	Schedule	Low (69/week)	70	Base
Large Hospital	Schedule	High (383/week)	400	Base
Equal Patient Mix	Schedule	Base	200	Equal

- Arrival Variability: "Schedule" indicates that patients arrive randomly, with different average numbers of patients showing up each hour based on a schedule. "Constant" means that a constant inter-arrival time is specified for each arrival stream.
- Arrival Volume and Number of Beds are conceptually combined as a Hospital Scale factor. Attempts to alter only arrivals or beds, without changing the other, led to a hospital that is either always nearly empty or always overly full and amassing more waiting patients.
- Patient Mix refers to the percent of each type of patient in the patient population. "Base" refers to the mix derived from aggregate data. "Equal" indicates that all patient types are equally likely to be present in the hospital.

Appendix IV: Attached Computer Files

The enclosed CD holds the computer files referenced in this thesis document. In the folder Simulation Model, the Park Ridge hospital simulation model (for Arena 7.01) is included. It includes numerous named views for navigation, simple animation, and several utilization plots and patient counter displays. In the folder Neural Networks are several workspace files (from MATLAB version 6.0.0, release 12 and MATLAB's Neural Network Toolbox version 4.0) that contain the networks used in these experiments:

- dug.mat – this is the one-hour expected value network; file contains the network "net"
- dug_6hr_nw.mat – this is the six-hour expected value network; file contains a network also named "net"